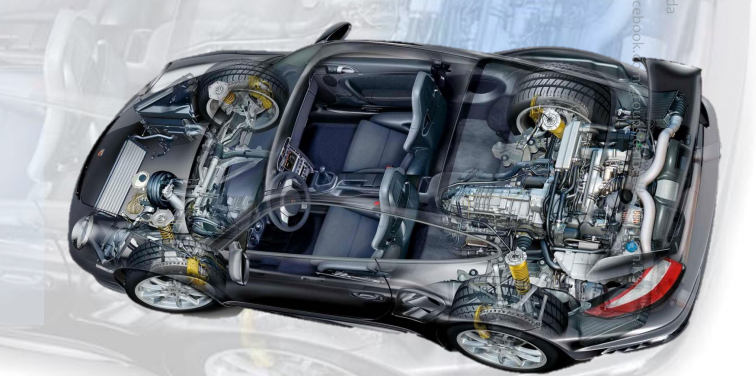
**Mastering Embedded System Online Diploma**

[www.learn-in-depth.com](https://www.learn-in-depth-store.com)



**Topic: First Term Project 2**

**Project: High Pressure detection.**

**Eng. Slim benhammouda**

**My profile:** [**progress page**](https://www.learn-in-depth-store.com/account/applystar)

**Contents :**

* Case study
* Methodology
* Requirement Diagram
* Space Exploration (HW/SW Partitioning)
* System Analysis: Use Case Diagram
* System Analysis: Activity Diagram
* System Analysis: Sequence Diagram
* System Design
* System Design Simulation.
* A brief look at source files
* Sections and symbols for final executable file
* Finding the entry point using readelf utility
* A brief look at the map file
* Hardware Simulation
* **case study:**

In order to protect the crew of a cabin from being exposed to high-pressure levels, we need to warn them when the pressure exceeds 20 bars by turning on an alarm for 60 seconds.

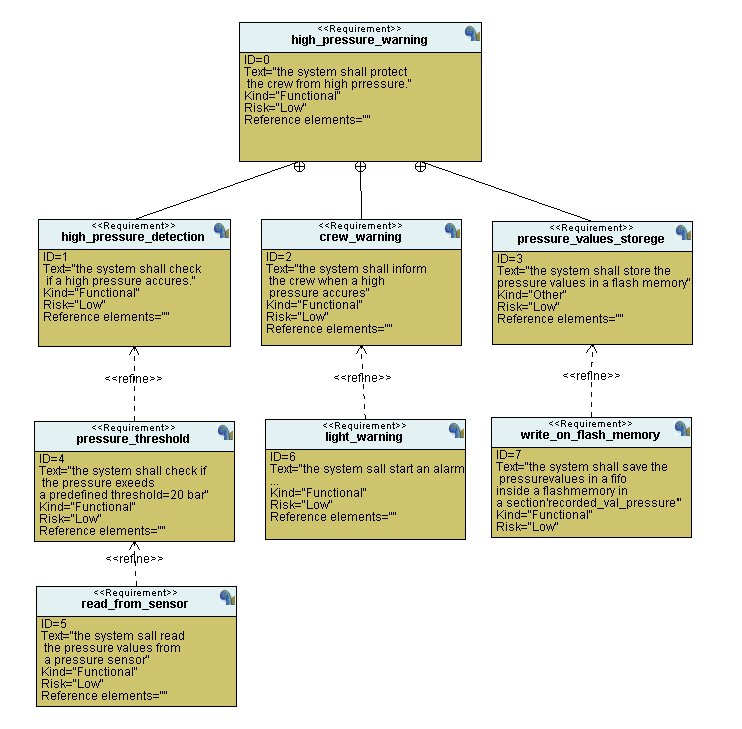
Keep track of the measured values by storing them in a flash disc (optional).

**Assumptions:**

* This system never faces a power cut.
* The pressure sensor never fails.
* The alarm actuator never fails.
* Maintenance issues are not modeled.

**Methodology:**

Since we have multiple modules to dial with, which have to be integrated and tested separately due to the project sensitivity .we choose to follow the agile methodology that offers a reliable model of testing.

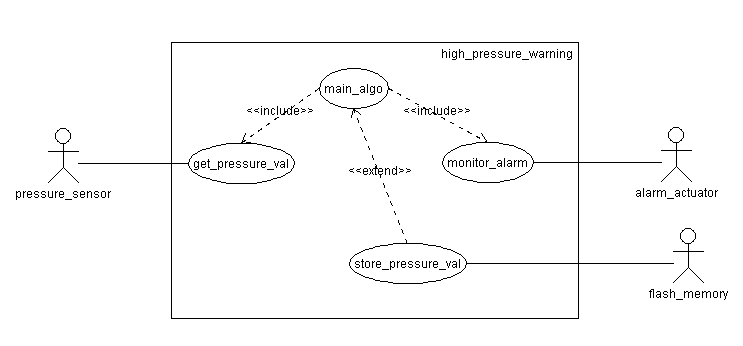
* **Requirement diagram: **

**-Space Exploration (HW/SW Partitioning):**

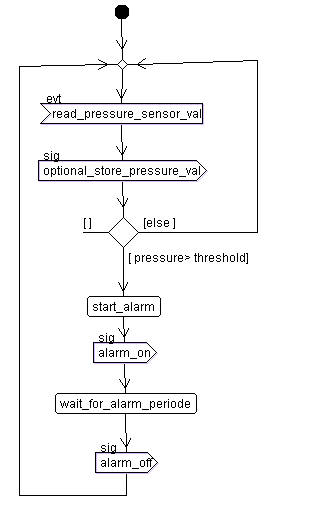
In this project, we choose to use the STM32 microcontroller with its cortex M3 CPU as an optimum solution for the hardware, which had shown an excellent performance in the exploration and partitioning part.

**- System analysais:**

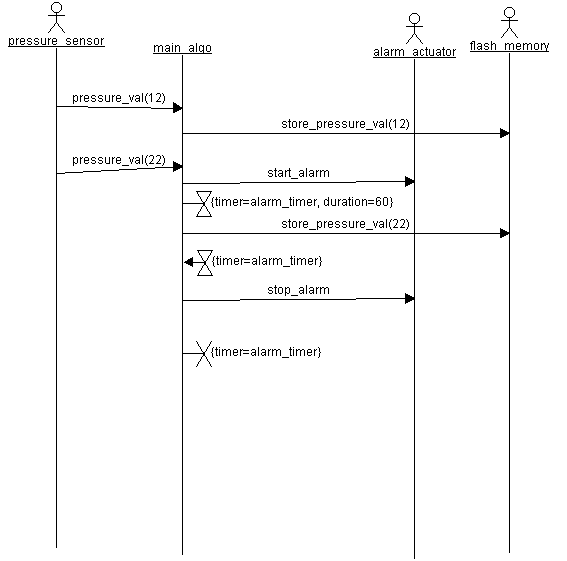
* **Use Case Diagram:**



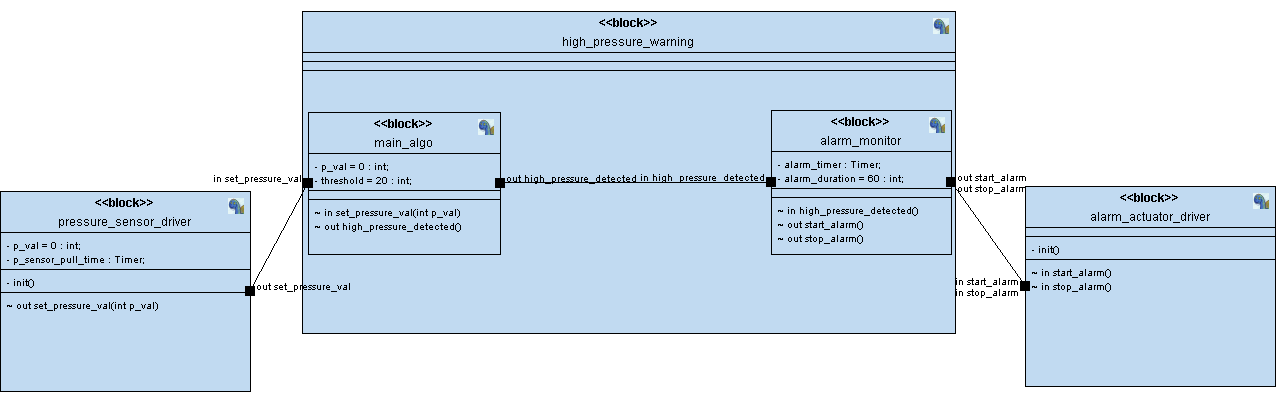
* **Activity Diagram:**

****

* **Sequence Diagram:**

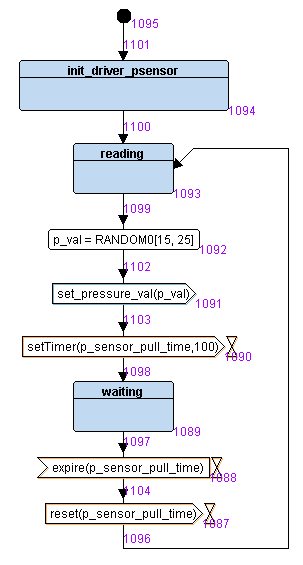


**-System design :**

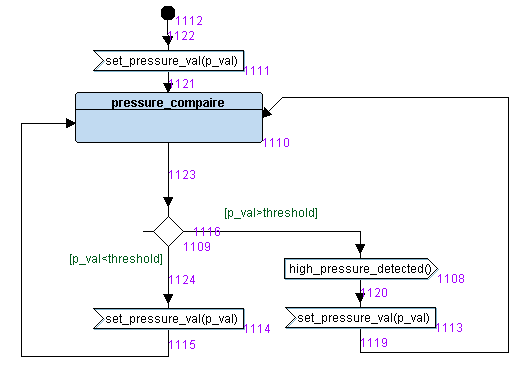


**Block diagram**

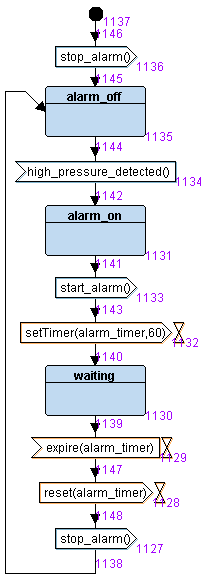
* **Pressure Sensor State Diagram :**

****

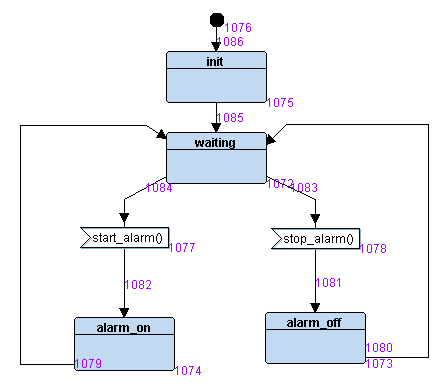
* **Main algorithm State Diagram :**

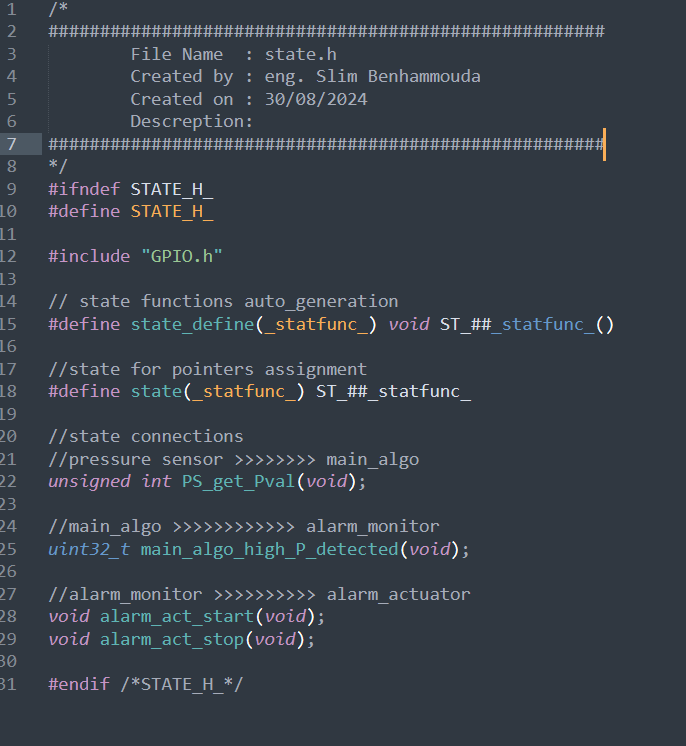
****

* **Alarm Monitor State Diagram:**

****

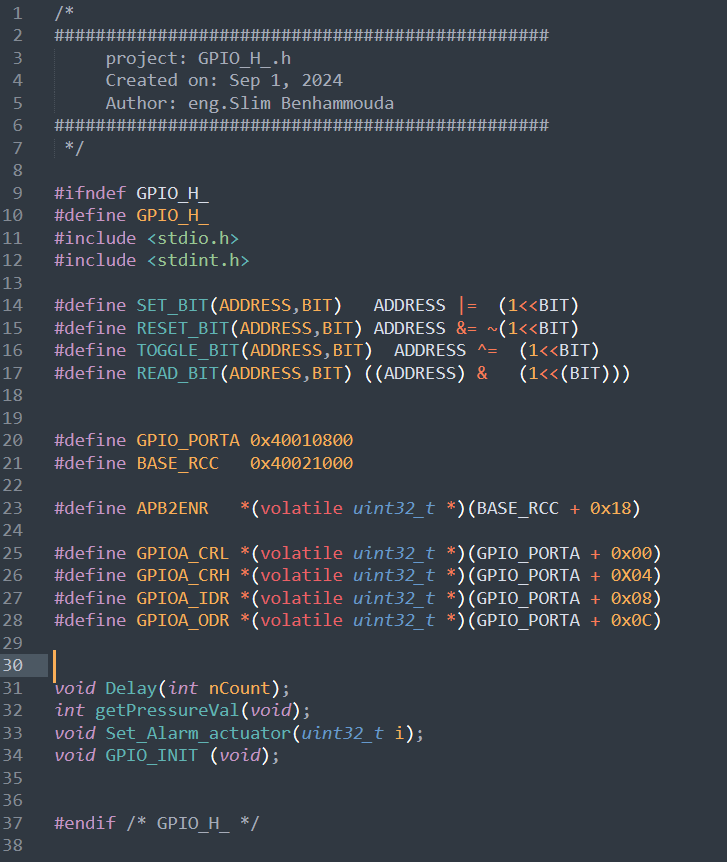
* **Alarm Actuator State Diagram:**

****

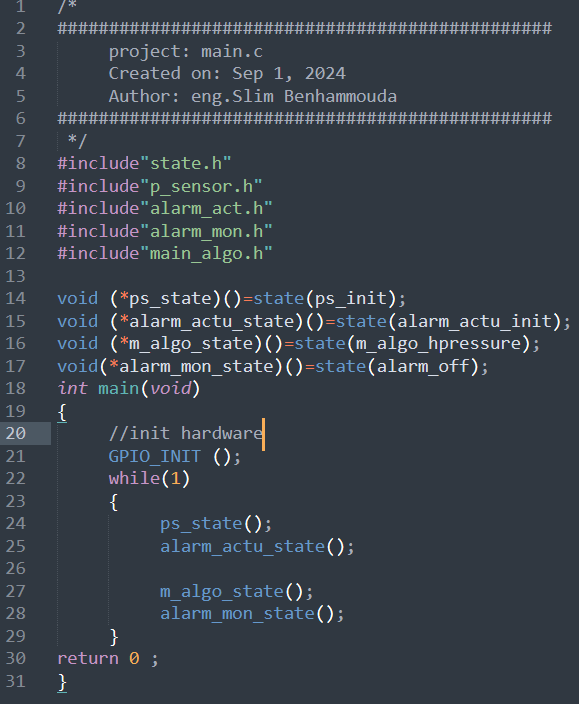
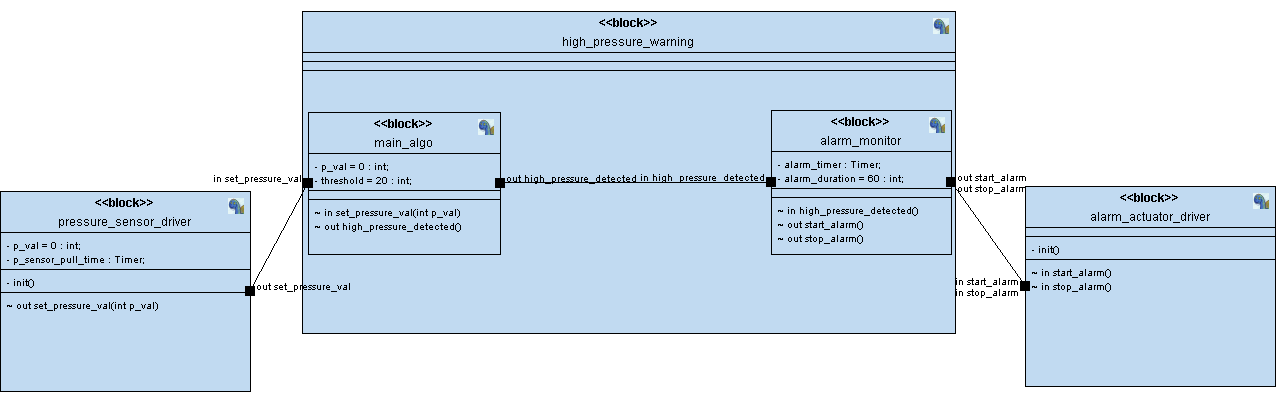
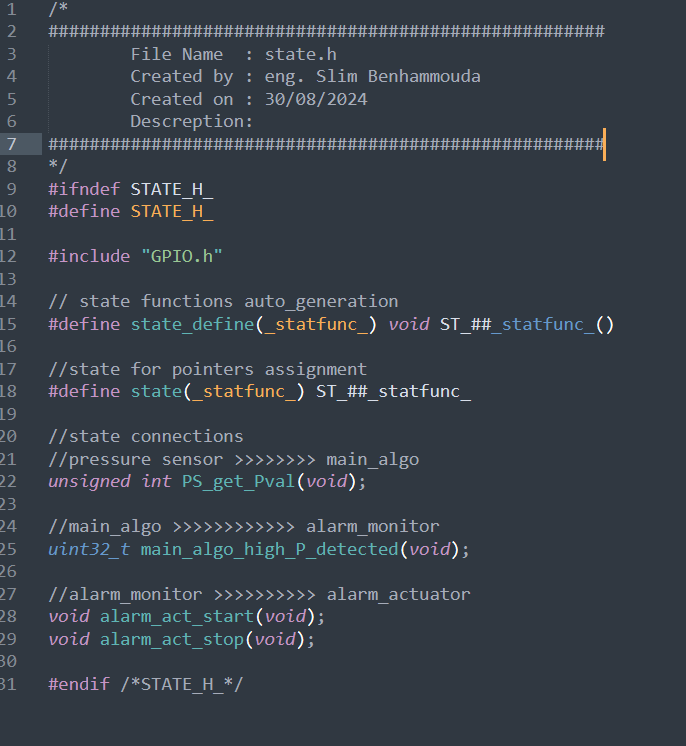
$

**Codes implementation:**

**Hardware driver : GPIO.C & GPIO.h**

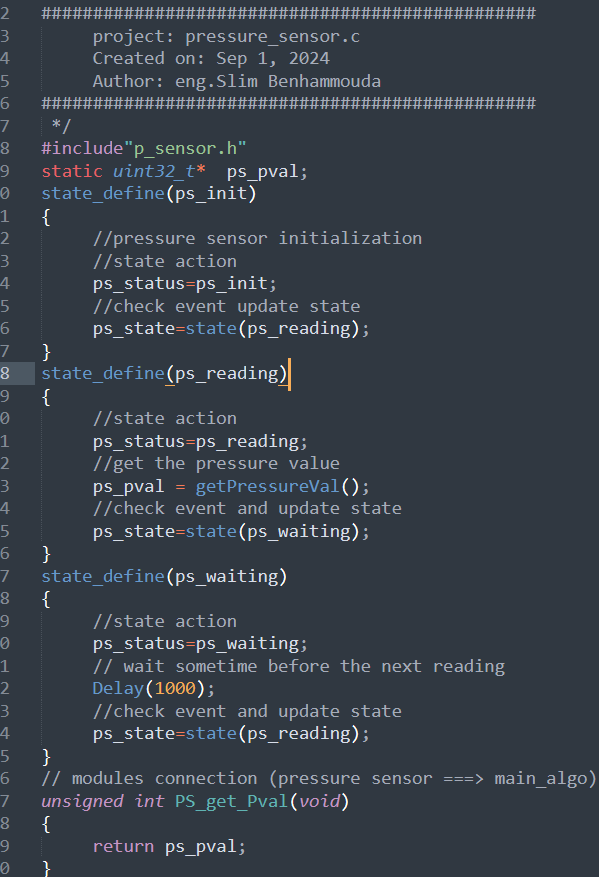
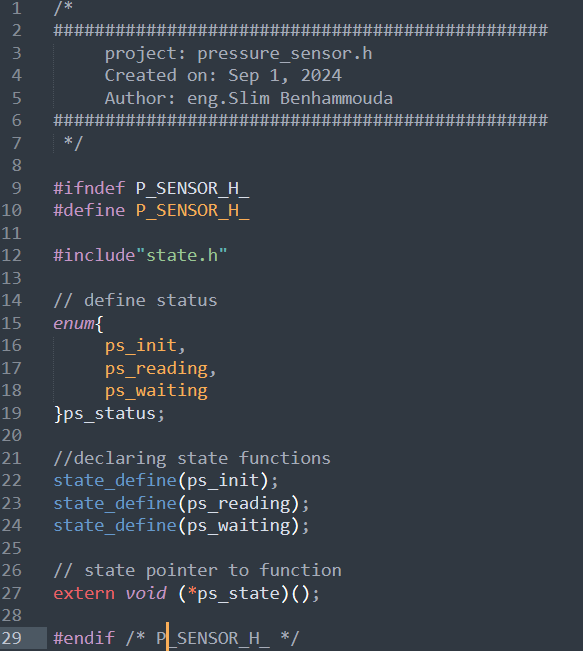
****

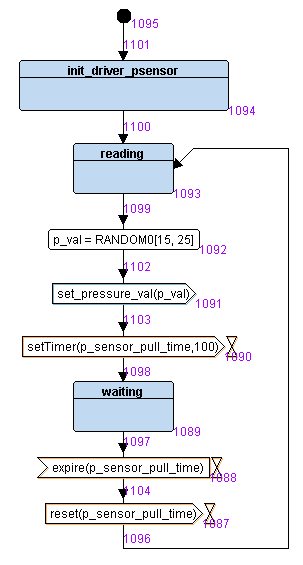
**Main.c: state.h:**

****

**Pressure sensor module:**

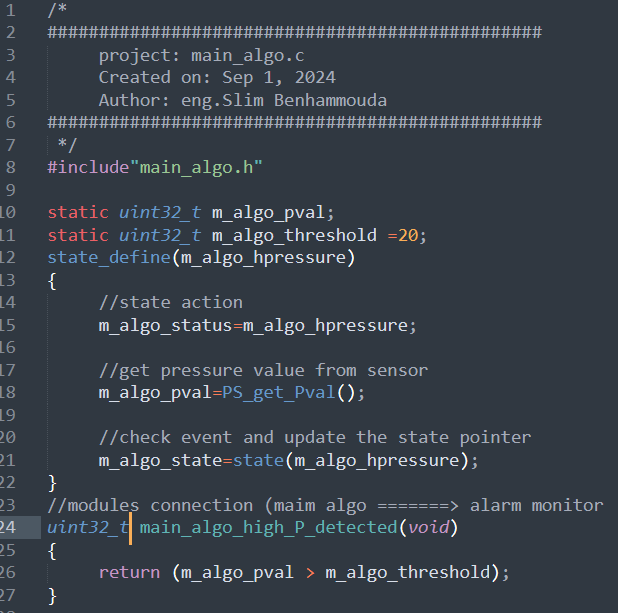
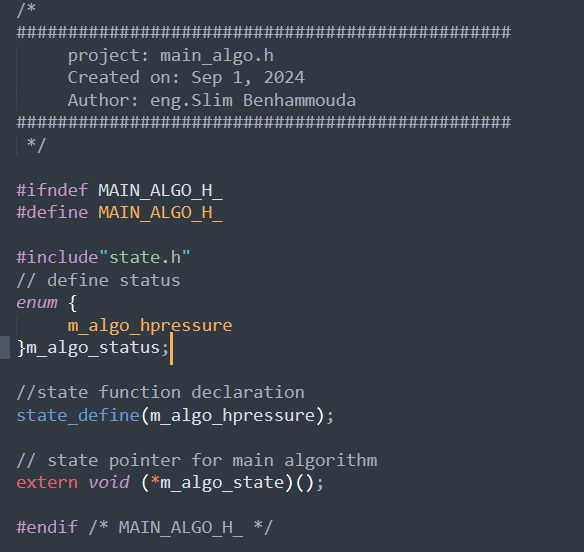
**Pressure\_sensor.h : Pressure\_sensor.c :**

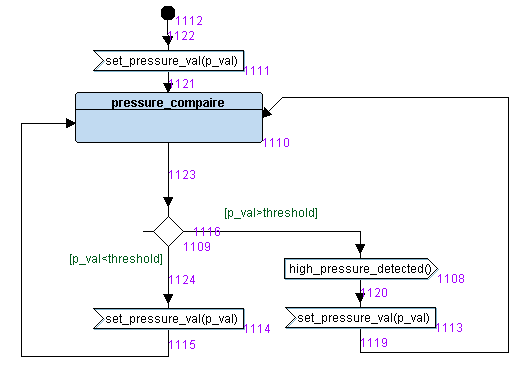
****

****

**Main algorithm module:**

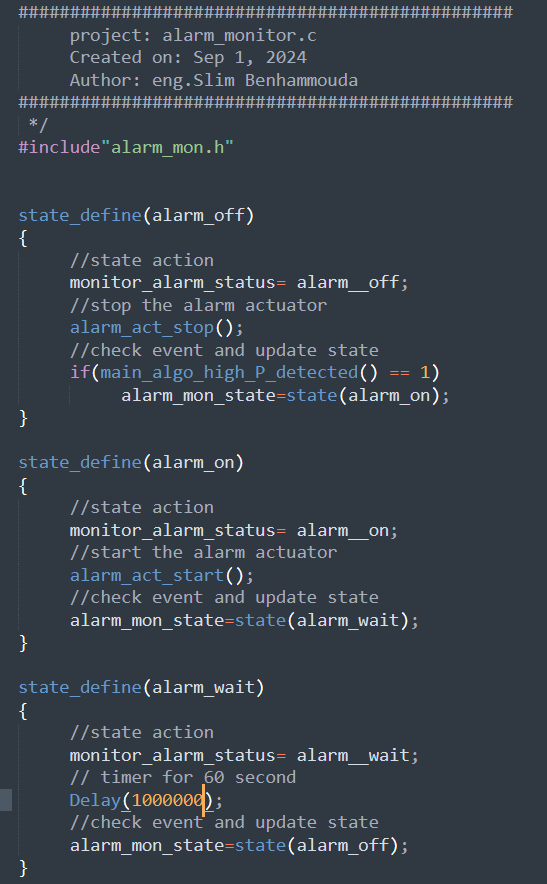
**Main\_algo.h : Main\_algo.c**

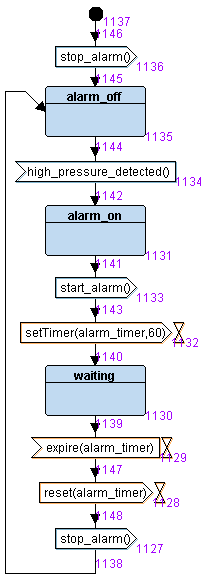
****

****

**Alarm monitor module:**

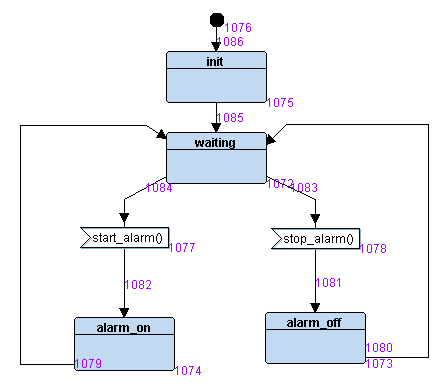
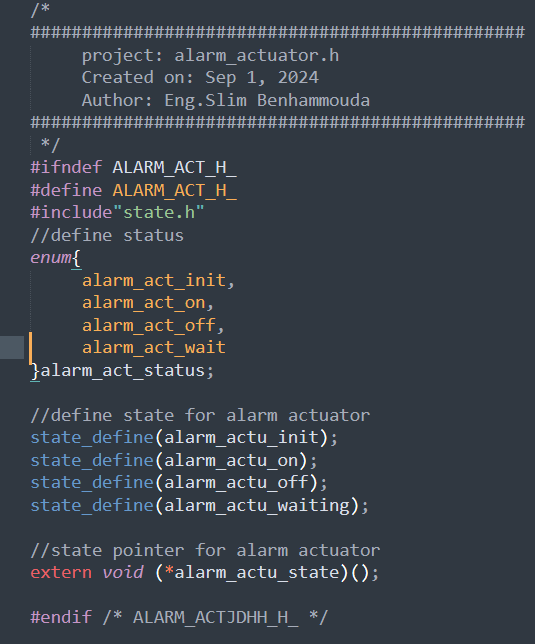
**Alarm\_monitor.h: Alarm\_monitor.c**

****

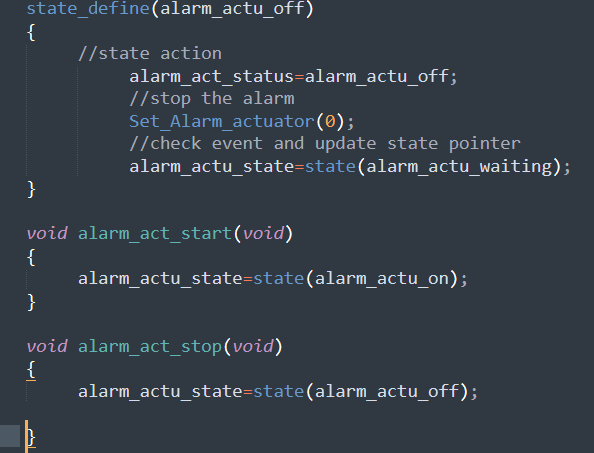
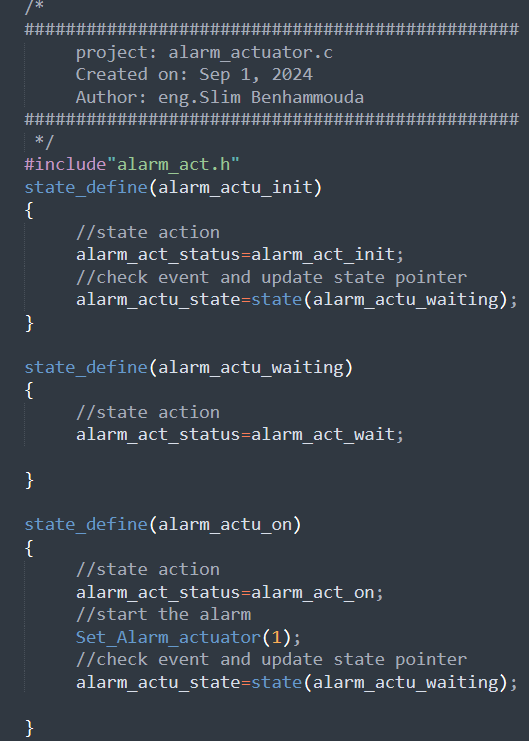
****

**Alarm actuator module:**

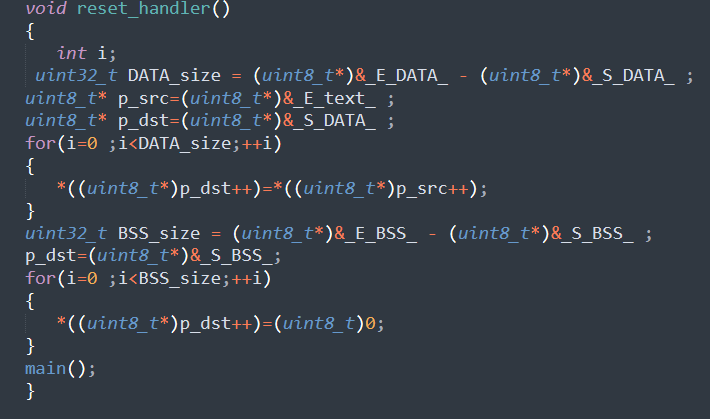
**Alarm\_actuator.h:**

****

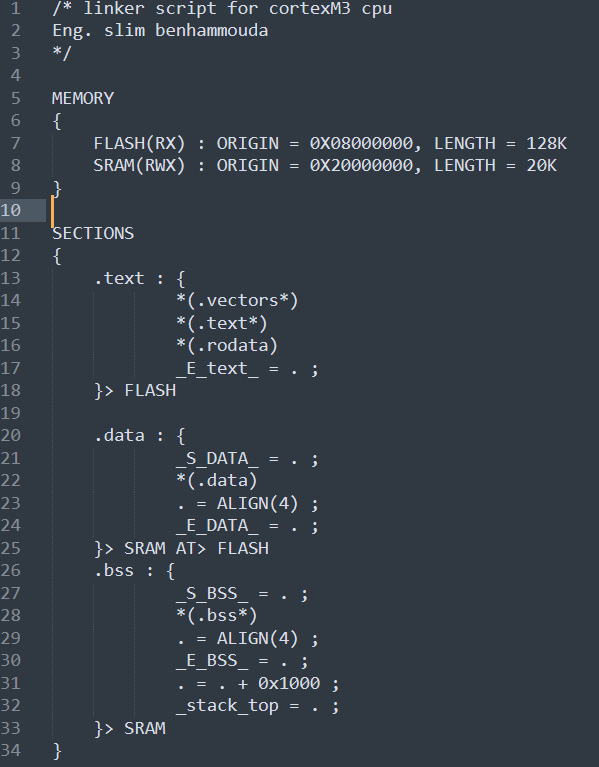
**Alarm\_actuator.c:**

****

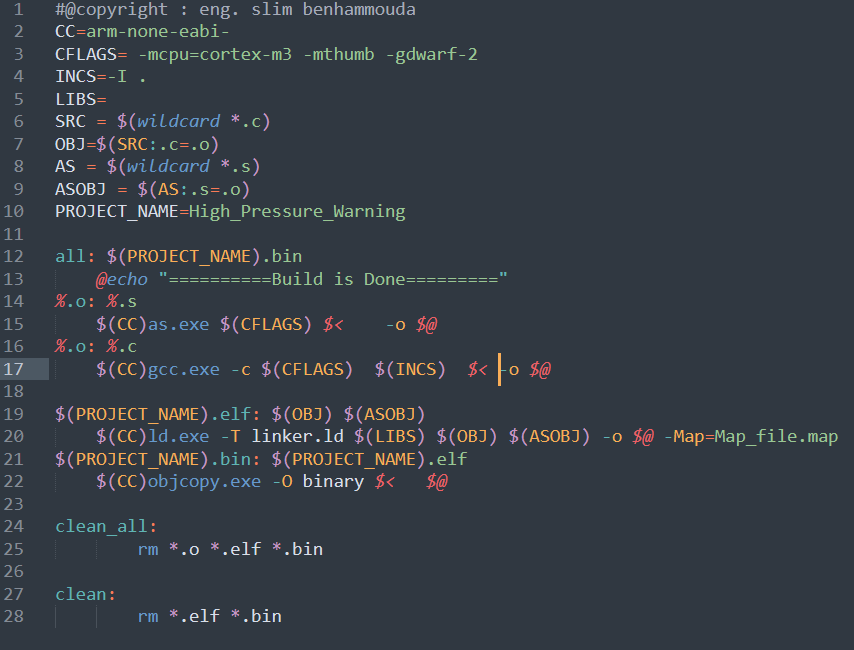
**Startup.c:**

****

**Linker\_script:**

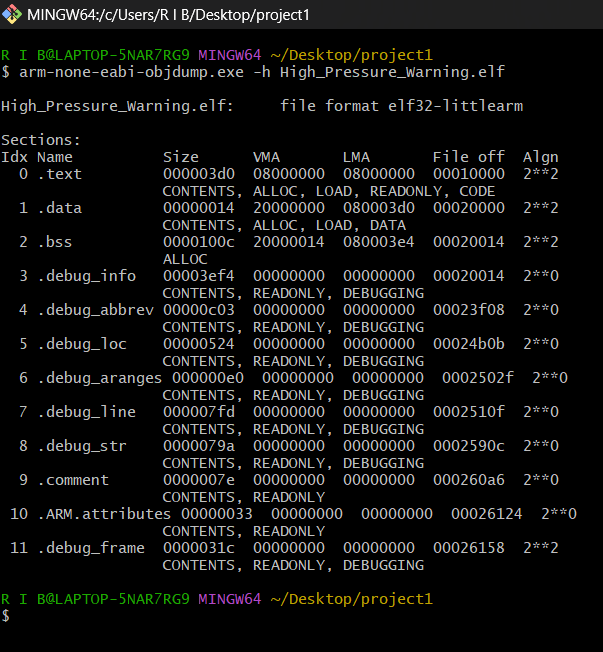
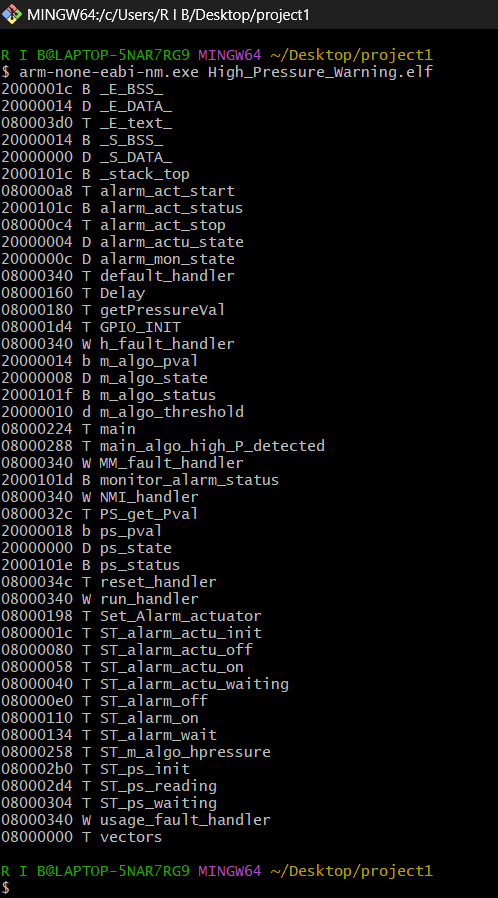
****

**Make file:**

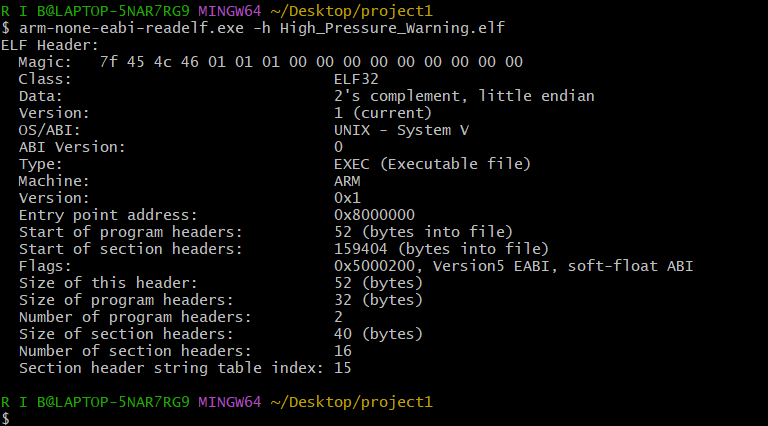
****

* **Sections and symbols for final executable**

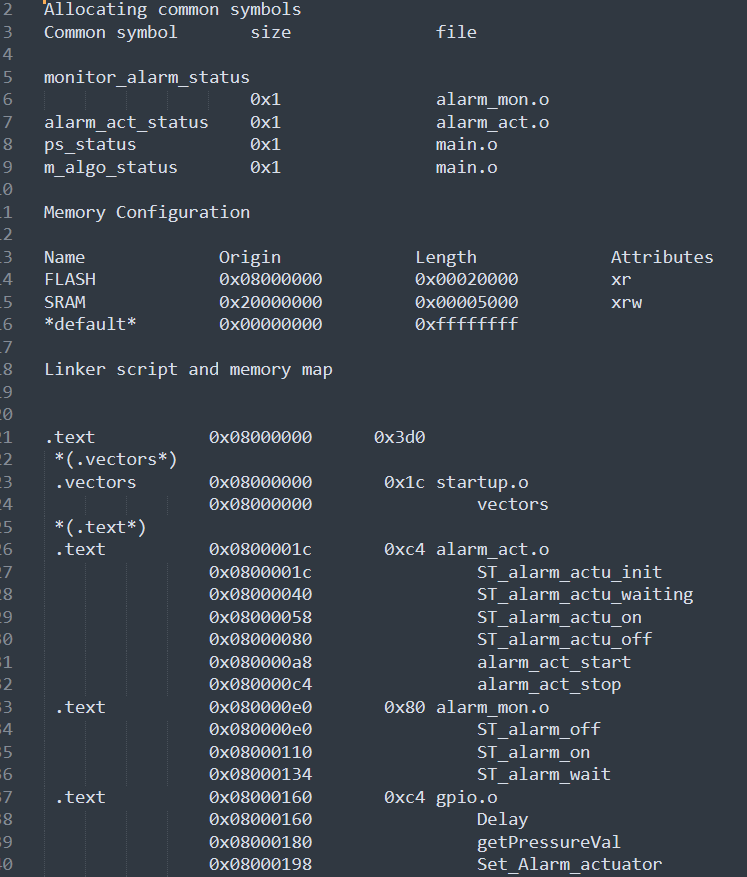
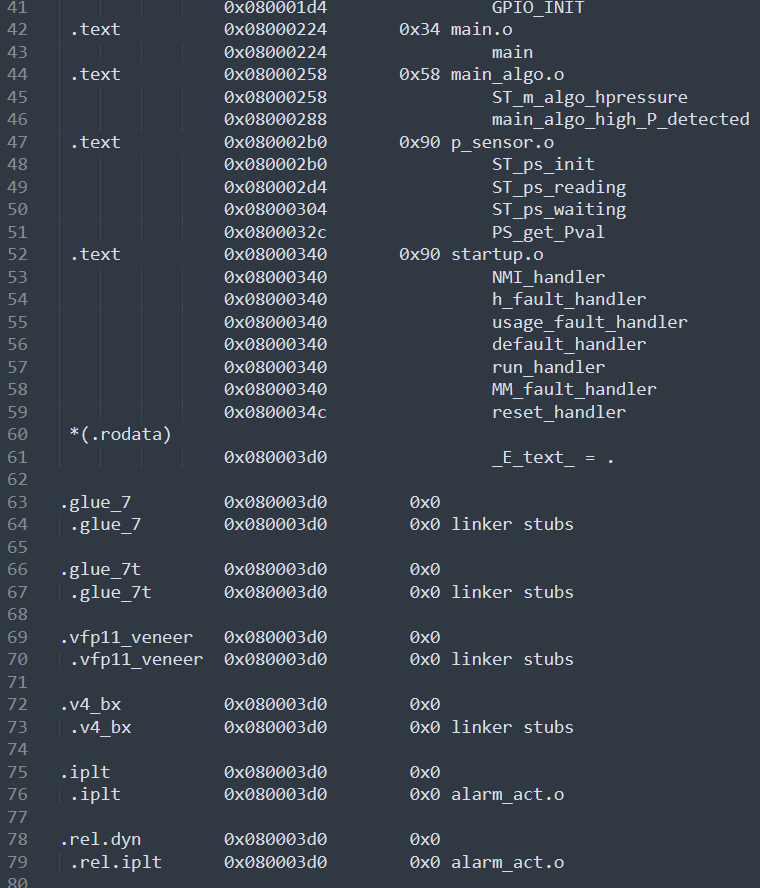
**Sections: Symbol table:**

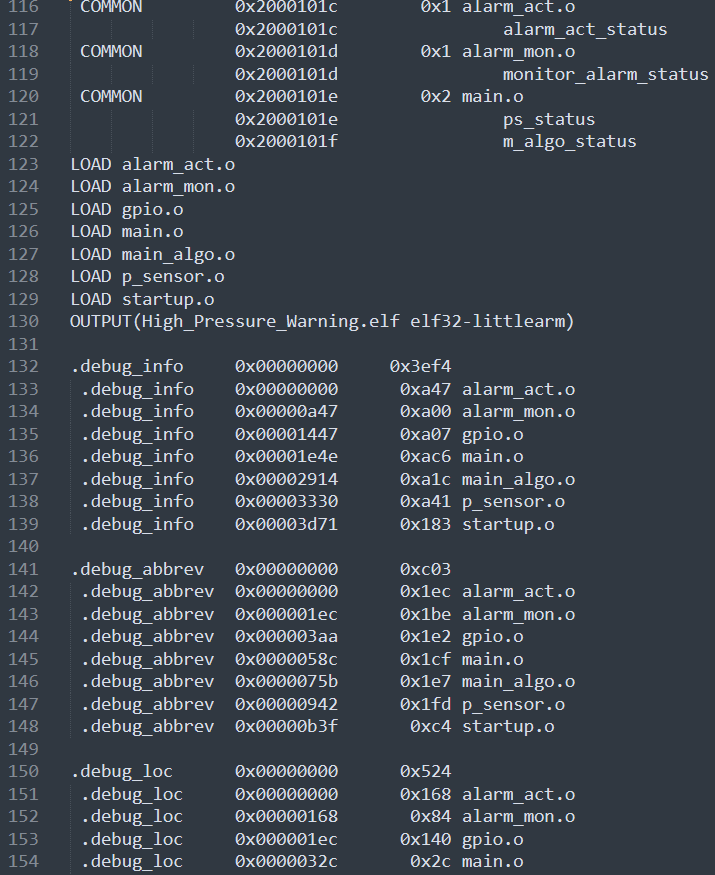
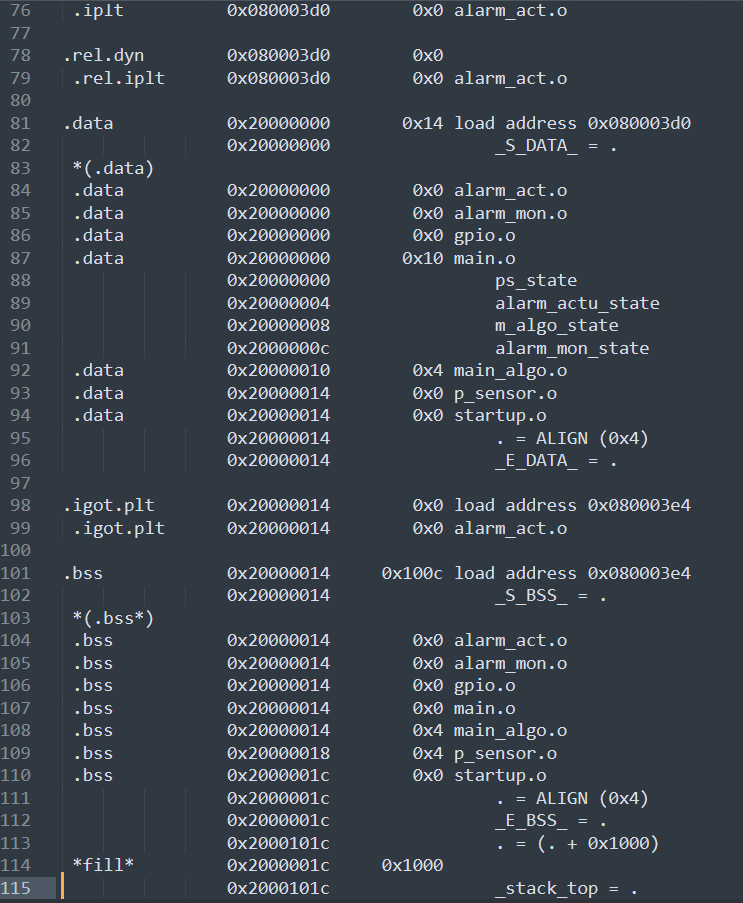
 ****

* **Finding the entry point using readelf utility:**



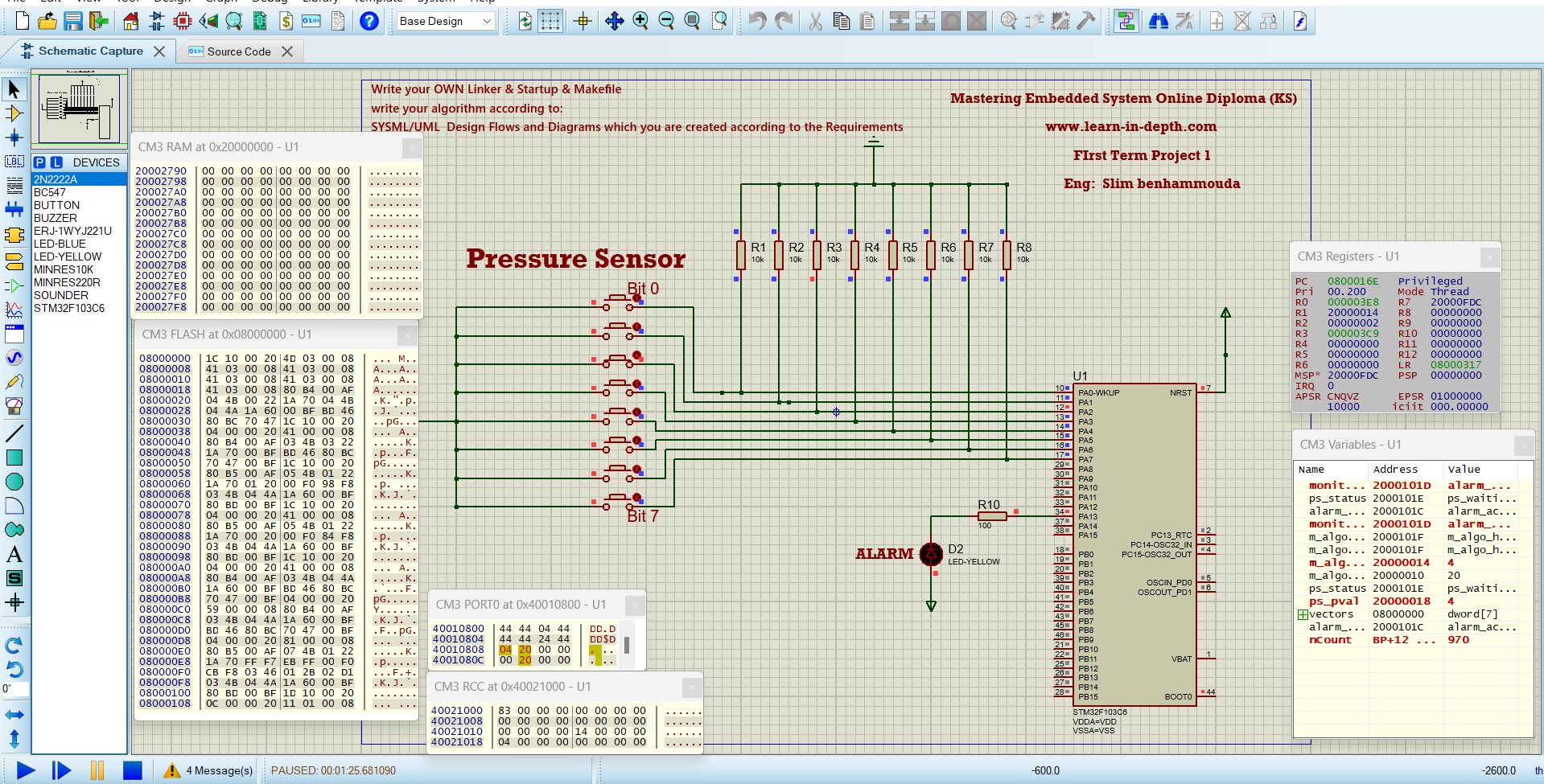
* A brief look at the map file:

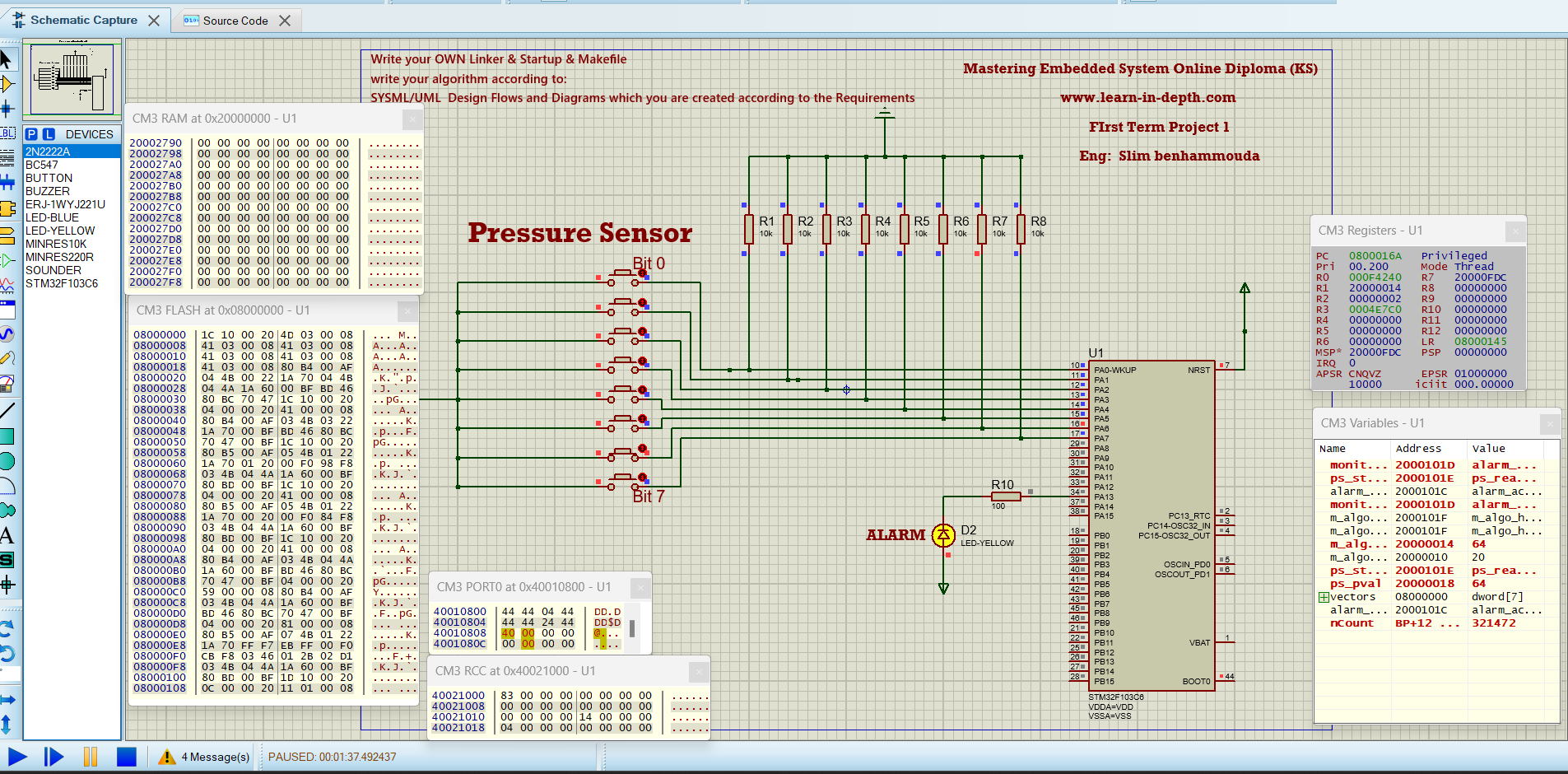


* **Hardware simulation :**

**P\_val = 4**

****

**P\_val = 64**

****